

We claim:

1. A method of allocating bandwidth of a limited bandwidth link to dataflows containing packets, including the steps of:

adaptively adjusting the number of buckets dependent upon the number of active dataflows, where each bucket has a number of tokens allocated to said bucket for use by the corresponding dataflow, said number of tokens dependent upon a weighted value for said corresponding dataflow, wherein queueing of said packets for utilization of said limited bandwidth link is dependent upon said tokens; and

adaptively reallocating tokens to one or more buckets in accordance with a weighted value for each of said dataflows.

2. A method according to claim 1, wherein the adaptive adjusting step further includes the step of creating an additional bucket for each additional dataflow, wherein the token-carrying capacity of said additional bucket is dependent upon a weighted value for said additional dataflow and said additional bucket is initially filled with tokens.

3. A method according to claim 1, wherein the adaptive adjusting step further includes the step of deleting a bucket when the dataflow corresponding to that bucket becomes inactive.

4. A method according to claim 3 further including the step of distributing the tokens from the deleted bucket amongst one or more of the other remaining buckets.

5. A method according to claim 1, further including the steps of:  
queueing one or more packets of a dataflow for utilization of said limited bandwidth link;

removing a number of tokens from the bucket corresponding to said dataflow, wherein said number of tokens is dependent upon the size of said one or more packets; and  
making said number of tokens available for reallocation.

6. A method according to claim 5, said method further including the step of:  
dropping one or more received packets of a dataflow when the bucket corresponding to

said dataflow has insufficient tokens for queueing of said one or more packets.

7. A method according to claim 5, further including the step of queueing received packets of diverse dataflows in a single queue.

8. A method according to claim 1 or claim 2, wherein two or more of said dataflows comprise heterogeneous dataflows.

9. A method according to claim 1 or claim 2, further including the steps of aggregating and treating two or more of said dataflows as a single dataflow.

10. A method according to claim 1 or claim 2, wherein one or more of said dataflows comprise hierarchical dataflows and each level of an hierarchical dataflow is treated as a single dataflow.

11. A method according to any one of the preceeding claims, wherein the total number of said tokens is conserved.

12. A method according to any one of the preceeding claims, wherein the rate of transmission of said packets across said limited bandwidth link is unaffected by the application of said method.

13. A system for allocating bandwidth of a limited bandwidth link to dataflows containing packets, including:

means for adaptively adjusting the number of buckets dependent upon the number of active dataflows, where each bucket has a number of tokens allocated to said bucket for use by the corresponding dataflow, said number of tokens dependent upon a weighted value for said corresponding dataflow, wherein queueing of said packets for utilization of said limited bandwidth link is dependent upon said tokens; and

means for adaptively reallocating tokens to one or more buckets in accordance with a weighted value for each of said dataflows.

14. A system according to claim 13, wherein the means for adaptively adjusting further includes means for creating an additional bucket for each additional dataflow, wherein the token-carrying capacity of said additional bucket is dependent upon a

weighted value for said additional dataflow and said additional bucket is initially filled with tokens.

15. A system according to claim 13, wherein the means for adaptively adjusting further includes means for deleting a bucket when the dataflow corresponding to that bucket becomes inactive.

16. A system according to claim 15, further including means for distributing the tokens from the deleted bucket amongst one or more of the other remaining buckets.

17. A system according to claim 13, further including:

means for queueing one or more packets of a dataflow for utilization of said limited bandwidth link;

means for removing a number of tokens from the bucket corresponding to said dataflow, wherein said number of tokens is dependent upon the size of said one or more packets; and

means for making said number of tokens available for reallocation.

18. A system according to claim 17, further including:

means for dropping one or more received packets of a dataflow when the bucket corresponding to said dataflow has insufficient tokens for queuing of said one or more packets.

19. A system according to claim 17, further including means for queuing received packets of diverse dataflows in a single queue.

20. A system according to claim 13 or claim 14, wherein two or more of said dataflows comprise heterogeneous dataflows.

21. A system according to claim 13 or claim 14, further including means for aggregating and treating two or more of said dataflows as a single dataflow.

22. A system according to claim 13 or claim 14 wherein one or more of said dataflows comprise hierarchical dataflows and each level of an hierarchical dataflow is treated as a single dataflow.

23. A system according to any one of claims 13 to 22, wherein the total number of said tokens is conserved.

24. A system according to any one of claims 13 to 23, wherein the rate of transmission of said packets across said limited bandwidth link is unaffected by the application of said system.

25. A computer program product including a computer readable medium with a computer program recorded therein for allocating bandwidth of a limited bandwidth link to dataflows containing packets, including:

computer program code means for adaptively adjusting the number of buckets dependent upon the number of active dataflows, where each bucket has a number of tokens allocated to said bucket for use by the corresponding dataflow, said number of tokens dependent upon a weighted value for said corresponding dataflow, wherein queueing of said packets for utilization of said limited bandwidth link is dependent upon said tokens; and

computer program code means for adaptively reallocating tokens to one or more buckets in accordance with a weighted value for each of said dataflows.

26. A computer program product according to claim 25, wherein the computer program code means for adaptively adjusting further includes computer program code means for creating an additional bucket for each additional dataflow, wherein the token-carrying capacity of said additional bucket is dependent upon a weighted value for said additional dataflow and said additional bucket is initially filled with tokens.

27. A computer program product according to claim 25, wherein the computer program code means for adaptively adjusting further includes computer program code means for deleting a bucket when the dataflow corresponding to that bucket becomes inactive.

28. A computer program product according to claim 27, further including computer program code means for distributing the tokens from the deleted bucket amongst one or more of the other remaining buckets.

29. A computer program according to claim 25, further including:

computer program code means for queuing one or more packets of a dataflow for utilization of said limited bandwidth link;

computer program code means for removing a number of tokens from the bucket corresponding to said dataflow, wherein said number of tokens is dependent upon the size of said one or more packets; and

computer program code means for making said number of tokens available for reallocation.

30. A computer program product according to claim 29, further including:

computer program code means for dropping one or more received packets of a dataflow when the bucket corresponding to said dataflow has insufficient tokens for queueing of said one or more packets.

31. A computer program product according to claim 29, further including computer program code means for queuing received packets of diverse dataflows in a single queue.

32. A computer program product according to claim 25 or claim 26, wherein two or more of said dataflows comprise heterogeneous dataflows.

33. A computer program product according to claim 25 or claim 26, further including computer program code means for aggregating and treating two or more of said dataflows as a single dataflow.

34. A computer program product according to claim 25 or claim 26, wherein one or more of said dataflows comprise hierarchical dataflows and each level of an hierarchical dataflow is treated as a single dataflow.

35. A computer program product according to any one of claims 25 to 34, wherein the total number of said tokens is conserved.

36. A computer program product according to any one of claims 25 to 35, wherein the rate of transmission of said packets across said limited bandwidth link is unaffected by the application of said computer program product.